Supplemental Information

Websites of sampled areas and their dominant vegetation:

Florida: http://ordway-swisher.ufl.edu. Sandhill community consisting of sparse longleaf pine/wiregrass savannah.

Hawai'i: http://www.hippnet.hawaii.edu. Laupāhoehoe Natural Area Reserve. *Metrosideros polymorpha* is the main canopy dominant, with very minor *Acacia koa.*

Alaska:

http://savanna.lternet.edu/site/research_site.php?site=bnz&research_site_id=252 Caribou-Poker Creek site. The vegetation is black spruce (*Picea mariana*) muskeg (acidic bog land with spaghum moss and reindeer lichen ground cover). http://www.lter.uaf.edu/bnz_vegetation.cfm for vegetation description of the region.

Utah: http://www.czen.org/content/domain-15-great-basin-onaqui-benmore. Extensive sagebrush steppe transitioning into juniper woodland.

nifH_tit reference set: A set of reference *nifH* sequences was constructed for use in FrameBot development as follows: We downloaded all available (1222) *nifH* bacterial isolates' protein and corresponding DNA sequences from the December 2011 release of the FunGene "nifH_tit" dataset, which contains only the *nifH* region amplified by the Poly *nifH* primers (internal to the primer binding region) (1). We clustered the protein sequences using the RDP mcClust program (http://fungene.cme.msu.edu) implementing the complete linkage algorithm. We

chose one reference sequence from each of the resulting 204 clusters formed at 90% protein identity and used these as the reference set for initial development and for testing the metric indexing strategy. This set contains representatives of all three *nifH* groups (I-III) and of the "*nifH*-like" sequences (group V-IV; Table S4). This reference set was used for *nifH* defined community analysis and for initial Metric Indexing development as described below.

Metric Indexing Speedup: For reading frame correction with *nifH* amplicon data, a small set of references can provide a reasonably close match to all query reads. However, we wanted to use a much larger reference set for assigning reads to nearest neighbors. Comparing each query to each member of a large reference set would slow FrameBot in proportion to the number of reference sequences. To speed up FrameBot, we applied the AESA metric indexing algorithm (2) in order to reduce the number of comparisons. This strategy involves pre-computing the edit distances between all pairs of reference sequences and making an initial estimate, Dm, of the maximum distance expected between members of the gene family (this estimate can be arbitrarily large with only a minor effect on performance). Then for a query Q, the distance between Q and a starting reference R (which may be chosen randomly), d(R,Q), is computed. If Dm > d(R,Q), Dm is replaced with d(R,Q). Only references with distances to R in the range of d(R,Q)+/-Dm are then retained, thus reducing the search space. A new R is chosen from among the remaining references and the process is repeated until only the closest match(es) remain.

For such a strategy, the distance measure used normally must meet the requirements for a metric distance, most importantly the triangle inequality. The distance between two sequences in a simple alignment (edit or Levenshtein distance) is metric. If different substitutions are given different costs, then the complex edit distance is metric only if the substitution costs are metric (3). It turns out that the Blosum matrices are not metric, but small adjustments to these matrices can be introduced to make them metric. We used such a modified Blosum 62 matrix when using FrameBot with a metric index (4). The resulting alignment score from FrameBot is then metric as long as there are no frameshifts, or as long as the frameshifts are in identical positions for all comparisons. However, it can easily be shown that changes in the position of inferred frameshifts can cause the resulting distances to violate the triangle inequality. We hypothesized that this violation would be rare in practice and tested this.

First we compared the results of using FrameBot with the standard Blosum 62 and metric Blosum 62 matrices (without index). For this we used the nifH_tit reference set. We selected 1000 sequences from the NEON dataset as a query set. For each scoring matrix, we exhaustively compared each query to all the reference sequences to find (one of) the closest match(es). Although in only 7% of trials was the closest match to a query the same for both matrices, in 91% of the trials, the nearest reference found using the two matrices had the same

percent amino acid identity with the query. In the remaining cases there was, on average, a difference of 1.3 amino acid mismatches between the nearest match found using the two matrices. As the difference sometimes favored each matrix, and as the difference was small, we considered the performance using the two matrices to be similar, as was previously concluded (4).

We then tested whether frameshifts would prevent the metric index from returning the closest match found by exhaustive comparison. In our initial trial, for 97.5% of the queries the indexing strategy returned (one of) the best match(es). For the remaining 2.5% of the queries, the indexing approach returned a more distant sequence, presumably because the inferred frameshift positions on some of the references chosen for comparison were not the same as those inferred in the comparison to the best match, hence leading to a violation of the triangle inequality. We reasoned that such a change in frameshift was more likely in comparisons with references distant from the query read, and that the errors were likely relatively small compared to this distance, so that by slightly increasing Dm at each step, we could avoid this problem. We randomly chose three sets of 1000 sequences from the NEON samples and determined that by slightly increasing the Dm used at each step by a fraction (0.2) of the distance d(RQ), we could avoid this problem for these test cases but still greatly reduce the number of comparisons. We then validated the performance using the augmented Zehr reference set on an additional randomly chosen subset of the NEON data (1000 reads). Fewer than 5% of the references, on average, were

compared to find the closest match and, for all 1000 trials, the same or an identically scoring closest match was found by exhaustive comparison, validating the indexing strategy.

***but* and *bph*A Amplicon Read Initial Processing:** Amplicon reads from two barcoded *bphA* samples were processed with the same parameters as for *nifH* except using the *bphA* defined community as reference for FrameBot. Amplicon reads from three barcoded *but* samples were processed with the same parameters as for *nifH* except using the *but* defined community as reference sets for FrameBot. Reads from the three *but* samples that passed the initial processing steps were separated into groups by closest matching defined community protein sequence using FrameBot. Only two groups of 698 reads in total from the two *Roseburia* strains were used for additional analysis because few reads were obtained from the other defined community members.

FragGeneScan and HMMFrame: FragGeneScan v.1.14 (5) was run with error model parameter matching the measured error rates as closely as possible, e.g., train_file_name “454_5” for the *nifH* and *bphA* defined community with error rates of about 0.5%, or “454_10” for the *but* defined community with error rates of about 1%. For some reason, FragGeneScan did not return the last amino acid for the vast majority of sequences in every dataset we tested. Since we believed this might be a minor programming error, we did not count this as a deletion error when comparing to the expected protein sequence. Also, since FragGeneScan

was originally developed as a gene finding tool, it sometimes, but not always, reported multiple partial sequences from a single query sequence when substitutions led to stop codons. This occurred, on average, less than 10 times per sample. These partial sequences from one query were concatenated prior to error analysis.

HMMFrame (6) was trained on three sets of sequences: the augmented Zehr reference set, a subset from Group I, II and III, and a subset from Group I only. HMMFrame was run using the default parameters. For the *but* gene, HMMFrame was trained on 10 high quality *but* sequences manually selected based on annotation and biochemical evidence. For the *bphA* gene HMMFrame was trained on 170 sequences selected from FunGene.

The *bphA* amplicon was too long for sequencing to read through to the reverse primer. The above quality filtering parameters were used except for reverse primer testing. All the reads shared at least 96% protein identity to one of the two defined community strains after frameshift correction with FrameBot.

AmpliconNoise: the *nifH* defined community reads were first filtered using the perl script “FlowsMinMax.pl” with primer input parameter “'ATCAGACACGTGCGA(C|T)CC(G|C)AA(A|G)GC(C|G|T)GACTC” (the 10 bp barcode and Poly *nifH* forward primer sequence), followed by the standard analysis pipeline: shell script test/run.sh. Both scripts are included with

AmpliconNoise v1.2 package (7). We found that as provided the shell script run.sh truncated all sequences to 220 bases. We modified the script to not truncate the sequences. The representative sequences returned by the AmpliconNoise were then processed by RDP Pyro Initial Process tool (8) only to remove barcode and forward primer. The reverse primer was not checked on these sequences since many had been truncated. These reads were subjected to frameshift-correction by FrameBot.

References:

1. **Poly F, Ranjard L, Nazaret S, Gourbière F, Monrozier LJ**. 2001. Comparison of nifH gene pools in soils and soil microenvironments with contrasting properties. Appl. Environ. Microbiol. **67**, 2255-2262.
2. **Vidal E.** 1986. An algorithm for finding nearest neighbours in (approximately) constant average time. Pattern Recognition Letters **4**, 145-157.
3. **Ristad ES, Yianilos PN.** 1998. Learning String-Edit Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**:522-532.
4. **Xu W**. 2006. On integrating biological sequence analysis with metric distance based database management systems. PhD thesis. University of Texas at Austin.
5. **Rho M, Tang H, Ye Y.** 2010. FragGeneScan: predicting genes in short and error-prone reads. Nucleic Acids Res. **38**:e191. doi:10.1093/nar/gkq747.

6. **Zhang Y, Sun Y.** 2011. HMM-FRAME: accurate protein domain classification for metagenomic sequences containing frameshift errors. BMC Bioinformatics **12**:198. doi:10.1186/1471-2105-12-198.
7. **Quince C, Lanzen A, Davenport RJ, Turnbaugh PJ**. 2011. Removing noise from pyrosequenced amplicons. BMC Bioinformatics **12**:38.
8. **Cole JR, Wang Q, Cardenas E, Fish J, Chai B, Farris RJ, Kulam-Syed-Mohideen AS, McGarrell DM, Marsh T, Garrity GM, Tiedje JM.** 2009. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. Nucleic Acids Res. **37** (Database issue):D141-145.